

Examenul de bacalaureat 2011
Proba E. d)
Proba scrisă la INFORMATICĂ
Limbajul C/C++

Varianta 6

Filiera teoretică, profilul real, specializarea științe ale naturii

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.
- În rezolvările cerute, identificatorii utilizați trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).

THEMA I

(30 Puncte)

Für Punkt 1 schreibt auf das Prüfungsblatt, den Buchstaben welcher der richtigen Antwort entspricht.

1. Sei unterstehende Sequenz:

```
if (x+y>6) if (x-y<3) cout<<1; | printf("1");  
else cout<<2; | printf("2");  
else cout<<3; | printf("3");
```

Die ganzen positiven Werte der Variablen **x** und **y** für die der Wert 2 angeschrieben wird, sind: **(4P.)**

- | | |
|------------------------------|------------------------------|
| a. x=1 und y=4 | b. x=4 und y=1 |
| c. x=5 und y=2 | d. x=5 und y=3 |

2. Sei nebenstehender Pseudocode-Algorithmus.

Man bezeichnet mit **$x \% y$** den Rest der Division der natürlichen Zahl **x** durch die natürliche, von Null verschiedene Zahl **y**.

- a. Schreibt die Zahl, nach dem Durchführen des Algorithmus die angeschrieben wird, wenn für die Variable **a** der Wert 6 eingelesen wird und für die Variable **b** der Wert 11. **(6P.)**
- b. Wenn für die Variable **a** der Wert 10 eingelesen wird, schreibt den Wert der für die Variable **b** eingelesen werden kann, so dass nach dem Durchführen des Algorithmus die Zahl 52 angeschrieben wird. **(4P.)**

```
lies a,b  
(natürliche, von Null verschiedene Zahlen,  $a \leq b$ )  
i ← a  
j ← b  
s ← 0  
solange i ≤ j wiederhole  
| s ← s + (i%2) * i + (j%2) * j  
| i ← i + 1  
| j ← j - 1  
■  
schreibe s
```

- c. Schreibt einen äquivalenten Pseudocode-Algorithmus der **KEINE** Kopfgesteuerten Wiederholungsstrukturen benutzen soll. **(6P.)**
- d. Schreibt das dem gegebenen Algorithmus entsprechende C/C++ Programm. **(10P.)**

THEMA II

(30 Puncte)

Für jeden der Punkte 1 und 2 schreibt auf das Prüfungsblatt den Buchstaben, welcher der richtigen Lösung entspricht.

1. Die Variable **x** ist von Typ **char**. Gebt an welcher C/C++ Ausdruck den Wert 1 hat wenn und nur wenn die Variable **x** den Buchstaben **A** speichert. (4P.)

a. **x==A** b. **x=='a'+1** c. **x=='B'-1** d. **x==char(A)**

2. Es seien die unterstehenden Ausdrücke die mit **E1** und **E2** bezeichnet wurden. In der nebenstehenden Sequenz sind alle Variablen von Typ ganz:
- | | |
|---------------------------------|---|
| E1: $d \leq n/2$ | <pre>ok=1; d=2; while(ok==1 &&) if(n%d==0) ok=0; else d=d+1;</pre> |
| E2: $d < \text{sqrt}(n)$ | |

Gebt den Ausdruck an, mit welchen die Auslassungspunkte ersetzt werden können, so dass, nach der Durchführung der erhaltenen Sequenz, die Variable **ok** den Wert 1 hat wenn und nur wenn die natürliche Zahl die in der Variablen **n** gespeichert ist, streng größer als 1 und Primzahl ist. (4P.)

- a. nur **E1** b. nur **E2**
c. **E1** oder **E2** d. keines der beiden Ausdrücke

Schreibt auf das Prüfungsblatt die Antwort für jede der folgenden Anforderungen.

3. Die ganzen Variablen **a**, **b** und **c** speichern die Längen der Seiten des Dreiecks **ABC**. Schreibt einen C/C++ Ausdruck der den Wert 1 hat wenn und nur wenn das Dreieck rechtwinklig ist. (6P.)
4. Es werden zwei natürliche Zahlen **x** und **y** (**x** hat unterschiedliche Ziffern, **x>9**) gelesen und es wird verlangt dass alle natürlichen Zahlen aufgeschrieben werden die kleiner als **y** sind und die aus **x** durch die Auslassung einer Ziffer und den Behalten der relativen Ordnung der anderen Ziffer, erhalten wird. Die Zahlen werden durch je ein Leerzeichen getrennt angeschrieben; wenn es keine solchen Zahlen gibt wird die Meldung **NU EXISTA** angeschrieben.
Beispiel: wenn **x=45216**, **y=4525** sind die angeschriebenen Werte, nicht unbedingt in dieser Ordnung **4216 4516 4521**.
- a) Schreibt im Pseudocode den Lösungsalgorithmus für die gestellte Aufgabe. (10P.)
b) Erläutert die Rolle aller Variablen, die in der Bearbeitung bei Punkt a) vorkommen und gebt die Eingabe- beziehungsweise die Ausgabedaten für die gestellte Aufgabe an. (6P.)

THEMA III

(30 Puncte)

Für Punkt 1 schreibt auf das Prüfungsblatt den Buchstaben, welcher der richtigen Lösung entspricht.

1. Es seien zwei eindimensionale Felder, das erste enthält Elemente die in steigender Reihenfolge gespeichert sind und das zweite Elemente die in fallender Reihenfolge gespeichert sind. Um einen Feld, mit fallend geordneten Elementen, aus beiden Felder, zu erhalten ist die Mischsortierung eine Methode die: **(4P.)**
 - a. zum richtigen Ergebnis führt, die aber im Bezug auf die Laufzeit nicht effizient ist
 - b. zum richtigen Ergebnis führt, und im Bezug auf die Laufzeit effizient ist
 - c. nur dann angewendet werden kann wenn, vor der Mischsortierung das erste Feld fallend geordnet wird
 - d. nur dann angewendet werden kann wenn, vor der Mischsortierung auch das zweite Feld steigend geordnet wird

Schreibt auf das Prüfungsblatt die Antwort für jede der folgenden Anforderungen.

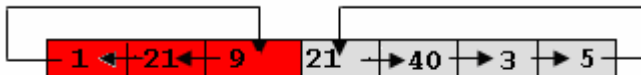
2. Es sei die Folge von natürlichen Zahlen in welcher die ersten drei Glieder $f_1=1$, $f_2=2$, $f_3=3$ sind. Das Element k ($k>3$) wird mit Hilfe der folgenden Formel berechnet:
 $f_k = f_{k-3} + 2 \cdot f_{k-1} - f_{k-2}$. In der nebenstehenden Sequenz sind alle Variablen von Typ ganz.
.....

```
for (i=4; i<=10; i++)  
{ d=c+2*a-b;  
  c=b; b=a; a=d;  
}
```

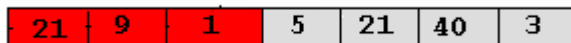
Schreibt die Anweisungen welche die Auslassungspunkte ersetzen können und welche die Variablen a , b und c so initialisieren dass, nach der Durchführung der erhaltenen Sequenz, die Variable d das 10. Glied der Folge speichert. **(6P.)**

3. Schreibt ein C/C++ Programm dass von der Tastatur zwei natürliche Zahlen n und k ($4 < n < 100$, $1 < k < n-1$) liest und dann die n Elemente eines eindimensionalen Feldes. Die Elemente sind natürliche Zahlen die höchstens vier Ziffern haben. Das Programm versetzt die ersten k Elemente des Feldes kreisförmig nach links, mit je einer Position und die restlichen Elemente kreisförmig nach rechts, mit je einer Position, wie im Beispiel. Dann schreibt das Programm, auf dem Bildschirm, die Elemente des veränderten Feldes, durch je ein Leerzeichen getrennt an.

Beispiel: wenn $n=7$, $k=3$ und das Feld v folgende Elemente hat:



dann ist das veränderte Feld:



(10P.)

4. Die Textdatei `bac.txt` hat auf einer einzigen Linie, durch je ein Leerzeichen getrennt, höchstens 10000000 natürliche Zahlen die höchstens 8 Ziffern haben. Die Datei enthält wenigstens zwei gerade Zahlen.

Es wird verlangt dass alle Zahlen aus der Datei `bac.txt` gelesen werden und auf dem Bildschirm die Summe der ersten und der letzten geraden Zahl angeschrieben wird. Man benutzt einen im Bezug auf die Laufzeit und den benötigten Speicher effizienten Algorithmus.

Beispiel: wenn die Datei die Zahlen 3 8 5 18 1 2 3 6 4 5 9 enthält, wird auf dem Bildschirm der Wert 12 ($12=8+4$) angeschrieben.

- a) Beschreibt in der Umgangssprache den benötigten Algorithmus und erklärt worin seine Effizienz besteht. **(4P.)**
- b) Schreibt das dem gegebenen Algorithmus entsprechende C/C++ Programm. **(6P.)**